«Исследование применимости алгоритмов сжатия данных к таблицам классификации в сетевом процессоре.»

Никифоров Никита Игоревич, 421 группа
Научные руководители:
Волканов Д. Ю., Скобцова Ю. А.

Актуальность

В данной работе рассматривается архитектура сетевого процессора (СП), в которой используется конвейерная архитектура. Конвейер состоит из последовательных вычислительных блоков, в каждом из которых находится независимое устройство памяти. В памяти вычислительного блока хранится программа классификации пакетов. Современные таблицы потоков занимают до нескольких десятков мегабайтов памяти [1]. В связи с малым объёмом памяти внутри одного вычислительного блока — 64 Кб, необходимо провести исследование существующих алгоритмов сжатия данных, и предложить их адаптацию для использования в рассматриваемом СП.

Не формальная постановка задачи

Необходимо исследовать применимость существующих алгоритмов сжатия данных в существующей архитектуре СП. Рассматриваемые алгоритмы должны удовлетворять следующим условиям:

- Размер итоговой таблицы потоков не должен превышать 512 Кб.
- Потери данных при использования алгоритмов сжатия не должны быть значительными.

Формальная постановка задачи

Введём формализацию OpenFlow таблиц. Упорядоченное множество всех рассматриваемых признаков в правилах обозначим $I = \{m_1, m_2, \dots, m_k\}$. Каждый признак m_i из множества признаков I характеризуется битовой строкой, некоторой длины $m_i \in \{0, 1, *\}_i^W$, в данном случае символ $*$ обозначает либой бит. При этом, если $\exists m_i^j \in m_i$, такое, что $m_i^j = *$, то для $\forall m_i^k$, где $k > j$, то $m_i^k = *$. Длиной признака обозначим $len(m_i) = W_i$

Представим таблицу потоков в виде множества правил $R = \{r_1, r_2, \dots, r_n\}$. С каждым правилом r_i связаны:

- номер i;
- приоритет $p_i \in Z_+$;
- вектор значений признаков $f_i = \{f_i^1, f_i^2, \dots, f_i^k\}$, где f_i^j соответствует значению признака $m_j \in I$.
- Набор инструкций, $A_i = \{a_1, a_2, \dots, a_z\}$, которые определяют дальнейшие действия сетевого процессора над пакетом.

Будем говорить, что заголовок пакета и его метаданные с вектором значений признаков $g = \{g^1, g^2, \dots, g^k\}$ соответствуют правилу $r_i \in R$ с вектором значений признаков $f_i = \{f_i^1, f_i^2, \dots, f_i^k\}$ и приоритетом p_i (правило $r_i \in R$ идентифицирует пакет с вектором знаений признаков g), если:

1. вектор значений признаков g соответствует вектору значений признаков f_i, то есть $\forall g_i \in g, len(g_i) = len(f_i)$. И $\forall f_i^{lj} \in f_i^l, f_i^{lj} \in \{*, g^{lj}\}, l = \overline{1, k}$;

2. приоритет p_i максимален среди всех правил $r_j \in R$, для которых g соответствует вектору значений признаков f_j.

Множество R также должно удовлетворять следующему ограничению. Для любых двух правил $r_i, r_j \in R, r_i \neq r_j$, если их вектора значений пересекаются, то есть существует набор значений признаков, который соответствует векторам значений признаков обоих правил, то $p_i \neq p_j$. Например, правила с векторами значений признаков $f_i = \{110, 011, 1*\}$ и $f_j = \{11*, 011, 11\}$ должны иметь разный приоритет, так как набор значений признаков $g = \{110, 011, 11\}$ соответствует обоим правилам.

Введём понятие анологичности множеств R_1 и R_2. Множество R_1 аналогично мноеству R_2, если для любого заголовка пакета, для которого существует идентифицурующее его правило $r_i \in R_1$, найдётся правило идентифицирующее его в множестве $r_j \in R_2$, при этом $A_i = A_j$.

Необходимо разработать алгоритм сжатия таблиц потоков, который будет переводить исходное множество $- R_1$, соответствующее исходной таблице потоков, в новое множество R_2, которое соответствует новой таблице потоков.

1. Множество R_1 должно быть анологично множеству R_2.

2. Мощность множества R_2 должна быть меньше либо равно мощности множества R_1.

Введём операцию последнего значащего бита признака $last(m_i) = j$, такое, что $m_i^j \in \{0, 1\}$ и $m_i^{(j+1)} = *$. Назовём правила $r_i \in R$ и $r_j \in R$ похожими, если для $\forall u \in len(f_i)$ верно, что $last(f_i^u) = last(f_j^u) = l$, при этом $f_i^{ul} \neq f_j^{ul}$, и $A_i = A_j$.

Список литературы

[1] Ori Rottenstreich и János Tapolcai. «Optimal rule caching and lossy compression for longest prefix matching». В: *IEEE/ACM Transactions on Networking* 25.2 (2016), с. 864—878.